universidad de león

Departamento de Matemáticas

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN CIBERSEGURIDAD

Trabajo de Fin de Máster

Análisis, diseño e implementación de una honeynet para el estudio de malware en entornos industriales IoT

Analysis, design, and implementation of a honeynet for the study of malware in industrial IoT environments

Autor: Ibai Castañón Osorio

Tutor Académico: Camino Fernández Llamas

Cotutor: Ricardo Julio Rodríguez Fernández

(Septiembre, 2021)

UNIVERSIDAD DE LEÓN

Departamento de Matemáticas

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN CIBERSEGURIDAD

Trabajo de Fin de Máster

ALUMNO: Ibai Castañón Osorio

TUTOR ACADÉMICO: Camino Fernández Llamas

COTUTOR: Ricardo Julio Rodríguez Fernández

TÍTULO: Análisis, diseño e implementación de una honeynet para el estudio de malware en entornos Industriales IoT

TITLE: Analysis, design and implementation of a honeynet for the study of malware in Industrial IoT environments

CONVOCATORIA: Septiembre, 2021

RESUMEN:

El uso de dispositivos *Internet of Things* (IoT) crece exponencialmente, esperándose que para el año 2025 haya un total de 21 billones de dispositivos IoT conectados. Dentro del paradigma IoT se encuentra un subconjunto orientado a procesos industriales denominados dispositivos *Industrial IoT* (IIoT) y que son una pieza clave de la Industria 4.0. A pesar de que estos dispositivos facilitan y potencian muchos aspectos de los procesos industriales, presentan numerosas vulnerabilidades que pueden llevar a consecuencias fatales. En este trabajo se propone un sistema compuesto por una honeynet basada en un entorno industrial IoT. Esta honeynet, desarrollada utilizando honeypots ya existentes de código abierto, se encarga de monitorizar los movimientos de la red y recoger las posibles muestras de software dañino que intenten desplegar los atacantes. Estas muestras son posteriormente mandadas a un sistema de análisis de malware para analizarlas y generar un informe. El objetivo de este trabajo es facilitar a los investigadores una herramienta que ayude en la recogida de registros de movimiento y muestras maliciosas para su estudio en entornos industriales IoT.

Palabras clave: Honeypot, IIoT, Malware, Malware analysis, ICS

Firma del alumno:	**VºBº Tutor:**

Agradecimientos

A mis compañeros, por el apoyo recibido durante estos dos años.

A mi familia, pareja y amigos, por su apoyo y comprensión.

Y a Ricardo J. Rodríguez, por guiarme y darme su apoyo para llevar a cabo este trabajo.

GRACIAS.

Abstract

The use of Internet of Things (IoT) devices is growing exponentially, with a total of 21 billion connected IoT devices expected by 2025. Within the IoT paradigm there is a subset oriented to industrial processes called Industrial IoT devices (IIoT) and which are a key piece of Industry 4.0. Although these devices facilitate and enhance many aspects of industrial processes, they present numerous vulnerabilities that can lead to fatal consequences. In this work, a system composed of a honeynet based on an industrial IoT environment is proposed. This honeynet, developed using existing open source honeypots, is responsible for monitoring network movements and collecting possible samples of malicious software that attackers try to deploy. These samples are later sent to a malware analysis system to analyze them and generate a report. The objective of this work is to provide researchers with a tool that helps in the collection of movement records and malicious samples for study in industrial IoT environments.

Índice general

Agradecimientos II

Abstract III

Índice de figuras VI

Índice de tablas VII

1. Introducción **1**

1.1. Objetivos 3

1.2. Estructura del trabajo 4

2. Conocimientos previos **5**

2.1. Sistemas de control industrial 5

2.2. Honeypot 8

2.3. Malware 10

3. Estado del arte **12**

3.1. Honeypots 12

3.1.1. Honeypots industriales 12

3.1.2. Honeypots en IoT 13

3.2. Malware 13

4. Análisis, diseño e implementación **15**

4.1. Análisis 15

4.1.1. Objetivo del sistema 15

4.1.2. Infraestructura 16

4.1.3. Gestión de información 19

4.2. Diseño 19

4.2.1. Arquitectura del sistema 19

4.2.2. Host 20
4.2.3. Contenerización 20
4.3. Tecnologías 23
4.4. Repositorio del sistema desarrollado 23

5. Pruebas de evaluación **24**
5.1. Escenarios de evaluación 24
5.2. Pruebas honeynet 25
5.2.1. IIoT Gateway 25
5.2.2. Honeyd - ICS 25
5.2.3. Cowrie - IoT Device 27
5.2.4. Problemas encontrados 28
5.3. Pruebas de sistema de análisis 28
5.3.1. Proceso 28
5.3.2. Problemas encontrados 29

6. Conclusiones y trabajo futuro **30**

Bibliografía **31**

A. Horas Invertidas **34**
A.1. Seguimiento 34
A.2. Desglose de horas 34

B. Instalación y configuración **36**
B.1. Requisitos previos 36
B.2. Instalación y configuración Honeynet 36
B.3. Instalación y configuración del sistema de análisis 38
B.4. Funcionamiento 39

C. Resultados **40**
C.1. Registro de logs 40
C.2. Reporte de análisis 42

Índice de figuras

1.1. Evolución dispositivos conectados. 2
1.2. Principales amenazas contra IoT en el año 2020. 2

2.1. Arquitectura ICS 6
2.2. Esquema sencillo de SCADA 7
2.3. Esquema Honeynet 9

4.1. Arquitectura IIoT Alto nivel 16
4.2. Esquema IIoT Gateway 17
4.3. Esquema ELK Stack 19
4.4. Esquema general del sistema 20
4.5. Arquitectura honeynet 21

5.1. Inicio de escritura de los ficheros logs de Honeyd 26
5.2. Captura de logs Nmap 26
5.3. Resultado del escaneo al host 26
5.4. Búsqueda de puertos abiertos en IoT 27

A.1. Horas invertidas por fase 35

B.1. Creación de patrón de indexado 37
B.2. Interfaz vboxnet0 creada 39

C.1. Muestra de logs de Honeyd. 40
C.2. Muestra de intentos de inicio de sesión en Cowrie. 41
C.3. Muestra de descargas de malware. 41
C.4. Información del fichero analizado. 42
C.5. Direcciones IP involucradas. 42
C.6. Comportamiento del malware. 43
C.7. Procesos que ejecuta el malware. 43

Índice de tablas

3.1. Resumen de las tres familias de malware IoT más características . . . 13

4.1. Comparativa de las honeypots industriales 18
4.2. Comparativa de las honeypots en IoT 18
4.3. Configuración de elementos en honeyd 22

5.1. Formato de log generado por honeyd 25

A.1. Horas invertidas por fase 35

Capítulo 1

Introducción

A día de hoy, gran cantidad de dispositivos de uso cotidiano como sensores de temperatura, puertas de garajes e incluso dispositivos *wearables* están conectados a Internet, es lo que se conoce como *Internet de las cosas* o *Internet of Things* (IoT). Los IoT nacen con el objetivo de hacer que los dispositivos sean inteligentes y capaces de actuar con autonomía, estando además conectados a la red de Internet. En base a este paradigma los sistemas de control industrial (ICS) han comenzado a incorporar dispositivos IoT dentro de su entorno dando lugar a lo que se conoce como *Industrial IoT* (IIoT). Los dispositivos IIoT son una de las piezas clave de la Industria 4.0 puesto que integran la parte de las tecnologías de información con las tecnologías de operación, haciendo que los procesos industriales sean más productivos. Se espera que esta tecnología tenga un fuerte impacto en la sociedad, llegando a ser un pilar fundamental de la Industria 4.0 [3]. En la Figura 1.1 se muestra el crecimiento estimado desde el año 2015 hasta el año 2025, que como puede observarse se estima un crecimiento anual del 10 %.

El uso de dispositivos IoT facilita muchos aspectos cotidianos tanto para las personas como para los procesos industriales. Sin embargo, estos dispositivos son vulnerables si no se utilizan con seguridad, algo que no es tenido en cuenta por gran cantidad de empresas a la hora de incorporarlos en sus infraestructuras.

Los principales objetivos de los ataques contra IoT son servidores, tráfico de redes, usuarios y los propios dispositivos [20], de lo que es importante destacar que gran parte de estos ataques tienen que ver con software dañino (*malware*).

La Figura 1.2 muestra el desglose de amenazas contra IoT durante el año 2020, diferenciándose tres grupos generales (de menor a mayor porcentaje): malas prácticas del usuario, exploits y el malware. Las amenazas de malware abarcan un 33 %, un número elevado teniendo en cuenta su peligrosidad. De estas amenazas, se destacan los malware de tipo gusano con un 12 %, seguidos de los ataques por ransomware con un 8 %. Más adelante (en la Sección 3.2) se explican en más detalle estos tipos de malware.

Los ataques de malware contra IoT cumplen con una anatomía que se puede definir en tres apartados [24]: *atacante*, *vector de ataque* y *fase de ataque*. Dentro de los vectores de ataque, destacan vectores tan comunes como contraseñas débiles, falta de cifrado, backdoors y la exposición a Internet.

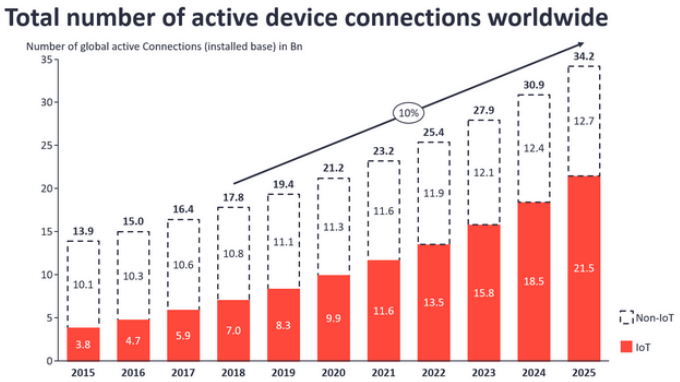

Figura 1.1: Evolución dispositivos conectados. Fuente: [23].

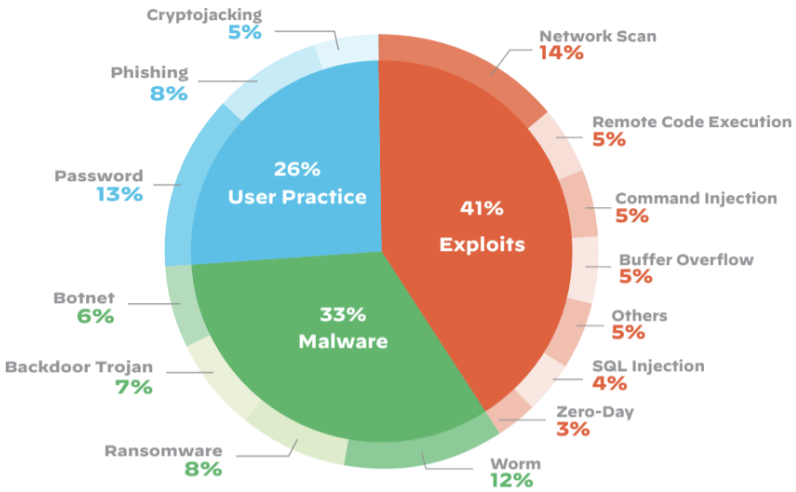

Figura 1.2: Principales amenazas contra IoT en el año 2020. Fuente: [26].

La fase de ataque se encuentra dividida en dos fases: *exploración y toma de control* y *lanzamiento del ataque*. En la primera fase, el atacante escanea direcciones IP en busca de puertos abiertos. Si el atacante encuentra puertos abiertos, intenta un inicio de sesión en el dispositivo utilizando combinaciones de usuario y contraseña comúnmente utilizados. En caso de conseguir acceso, la infección continúa hasta hacerse con el control del dispositivo para posteriormente pasar a la segunda fase, el lanzamiento del ataque.

El malware es un peligro para todos los dispositivos, y en IoT no ha sido para menos puesto que ha llegado a generar algunas situaciones críticas como en el caso de Mirai. Mirai es un malware que crea una botnet de dispositivos IoT y que en septiembre de 2016 se utilizó para realizar un ataque de denegación de servicio distribuido (DDoS, del inglés *Distributed Denial of Service*) contra el proveedor de DNS Dyn, y que dio lugar a la caída de plataformas importantes como GitHub, Netflix, Twitter y Reddit, entre otros [21].

El esperado incremento de los dispositivos IoT, junto con las consecuencias fatales que pueden traer los ataques a estos dispositivos, lleva a pensar en utilizar mecanismos de defensa que permitan mitigar estos ataques e investigar su funcionamiento para protegerse frente a ellos. Para ello, se puede recurrir al uso de honeypots como mecanismos de defensa y recogida de información. Una honeypot es un sistema que actúa como señuelo para atraer la atención de atacantes, simulando servicios reales susceptibles de ser atacados. Un buen diseño a la hora de crear una honeypot debe cumplir con una serie de requisitos para conseguir engañar a los atacantes y que comprometan el sistema simulado.

En el presente trabajo, se analizan los honeypots existentes y tipos de malware contra IoT para llevar a cabo el diseño de una honeynet, que es un sistema de alta interacción compuesto por varias honeypots y que trata de emular una red completa para obtener gran cantidad de información del atacante. El objetivo de esta honeynet es monitorizar movimientos a nivel de red, extrayendo todos aquellos elementos que hayan sido descargados por un atacante dentro de la red y que puedan ser sospechosos de ser maliciosos.

1.1. Objetivos

El objetivo general de este trabajo consiste en desarrollar una honeynet integrable dentro de un entorno industrial basado en IIoT para el estudio de malware.

Los objetivos específicos del trabajo y que llevarán a la consecución del objetivo general son los siguientes:

1. Analizar diferentes arquitecturas de los sistemas de control industrial y las honeypots existentes en la actualidad que mejor se adaptan a estos sistemas.
2. Diseñar una honeynet, en base a la información obtenida del análisis, para que emule una infraestructura industrial lo más realista posible.
3. Implementar el diseño obtenido en las fases anteriores.
4. Realizar pruebas sobre esta honeynet para comprobar su funcionamiento mediante la recogida de información e interacción con la misma y analizar las muestras obtenidas.

1.2. Estructura del trabajo

La estructura que sigue esta memoria es la siguiente. En el capítulo 2 se redactan los conocimientos previos, necesarios para que el lector tenga un correcto seguimiento del trabajo. En el capítulo 3 se realiza un estado del arte analizando tanto las herramientas como la literatura existentes en materia de honeypots. El capítulo 4 explica la fase de análisis, el diseño del sistema y las herramientas y tecnologías utilizadas. En el capítulo 5 se detallan las pruebas realizadas y la discusión de resultados, por último, en el capítulo 6 se presentan las conclusiones y el trabajo futuro.

Adicionalmente y para resolver algunas cuestiones desarrolladas a lo largo del documento, al final del trabajo se han añadido los siguientes anexos: el anexo A muestra las horas invertidas en el desarrollo; el anexo B muestra las instrucciones de instalación y el funcionamiento del sistema desarrollado y por último, el anexo C muestra los resultados obtenidos en el capítulo 5

Capítulo 2

Conocimientos previos

En este capítulo se describen los conocimientos previos de especial relevancia para una buena comprensión por parte del lector sobre el trabajo desarrollado. En primer lugar, se habla sobre los sistemas de control industrial y los diferentes componentes que los conforman. A continuación se explican las honeypot, desde qué son hasta qué tipos existen y cómo interactúan con los atacantes. Por último, se habla sobre malware y se explican los tipos de análisis que se hacen sobre este.

2.1. Sistemas de control industrial

El término *sistemas de control industrial* (ICS) engloba varios tipos de sistemas [25]. Dentro de este término se incluyen los sistemas de control de supervisión y adquisición de datos (SCADA), los sistemas de control distribuido (DCS), controladores de lógica programables (PLC) e interfaces hombre-máquina (HMI), entre otros. Todos estos sistemas combinados se pueden encontrar tanto en entornos industriales comunes como en infraestructuras críticas.

La arquitectura de los sistemas modernos de control industrial se compone básicamente de tres capas o niveles [10]:

- **Capa de gestión:** Esta capa es la encargada de la gestión de los sistemas de información como los MIS, ERP y MES. Todos estos sistemas son sistemas de organización industrial utilizados en la parte de gestión empresarial para control y gestión de procesos y recursos.
- **Capa de supervisión:** La capa de supervisión se encarga de la monitorización de procesos y gestión de bases de datos en tiempo real. Además, es la encargada

de comunicar la capa de campo con la de gestión y del control de las operaciones realizadas por los dispositivos a nivel de campo.

- **Capa de campo:** Este nivel es el que contiene todos los dispositivos como sensores, actuadores, transmisores y otros elementos. Es la principal responsable de toda la recogida de datos y de manipulación de los dispositivos partícipes en el proceso industrial.

En la Figura 2.1 se puede apreciar la arquitectura de tres niveles mencionada en el anterior párrafo.

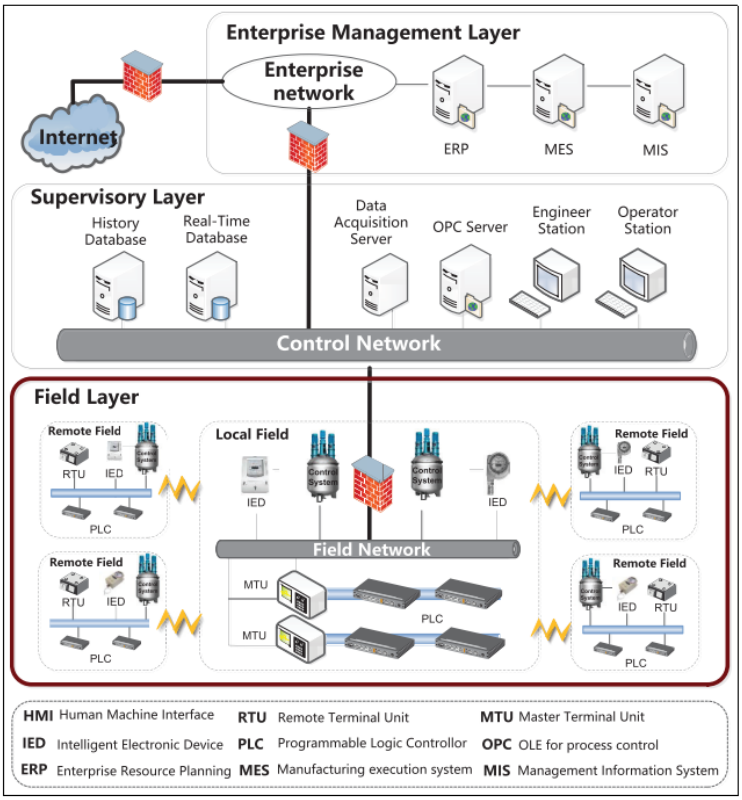

Figura 2.1: Arquitectura ICS. Fuente: [10].

A continuación, se describen algunos de los elementos más destacables de un ICS y que más interesantes pueden resultar para el trabajo desarrollado:

- **SCADA:** Dentro de los procesos industriales, la telemetría es a menudo necesaria para conectar diferentes dispositivos que se encuentren en distancias largas y así enviar comandos, programas o recibir información [18]. Los sistemas SCADA combinan la telemetría junto a la adquisición de datos y de esta forma ayudan a mantener el control de los procesos industriales. Mediante las interfaces hombre-máquina (HMI), los SCADA muestran la información y per-

miten a operadores capacitados interactuar con los elementos correspondientes a la capa de campo.

- **DCS:** Los sistemas de control distribuido se utilizan para el control de procesos industriales. Consta de elementos de control que se distribuyen de manera geográfica en la planta o área de control, a diferencia de los sistemas SCADA, que centralizan la información en una máquina central.
- **PLC:** Los PLC son elementos utilizados en procesos industriales de donde se requieran procesos de maniobra, control y señalización. Poseen una memoria programable por el usuario para almacenar instrucciones con el fin de implementar funciones específicas como el control de entrada/salida, temporización, comunicaciones e incluso procesamiento de datos y archivos [25].

La Figura 2.2 muestra un esquema sencillo de un sistema SCADA.

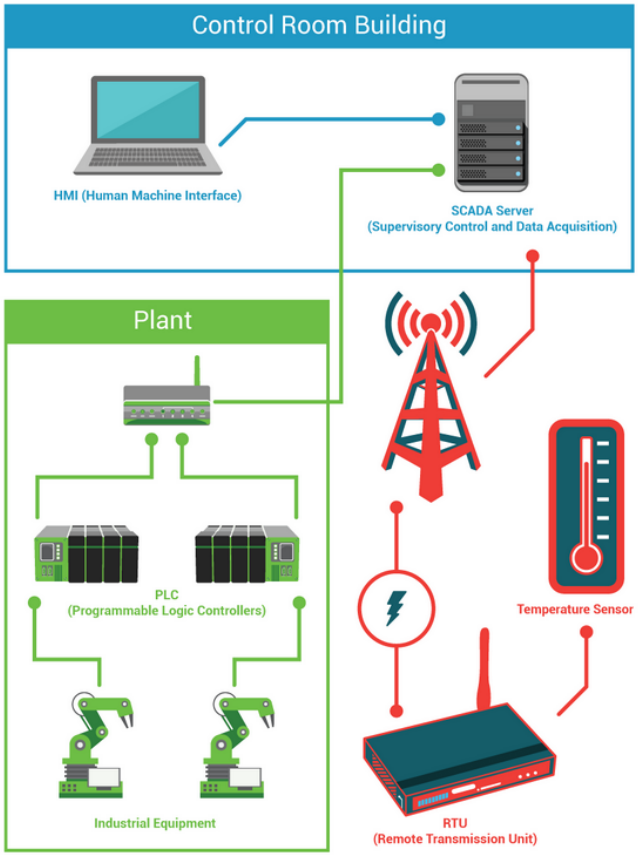

Figura 2.2: Esquema sencillo de SCADA. Fuente: [12].

2.2. Honeypot

Una honeypot es un sistema trampa creado con la intención de atraer a los atacantes y así evitar ataques contra infraestructuras o sistemas. Además de evitar los ataques, las honeypots se utilizan para recoger información sobre cualquier actividad maliciosa. Esta información puede ir desde hora y fecha del ataque, dirección IP o sistema operativo del atacante, hasta diccionarios utilizados para fuerza bruta. También se pueden extraer exploits, que son piezas de código que se aprovechan de vulnerabilidades para provocar un comportamiento no deseado, e incluso comandos usados para infiltrarse en el sistema [7].

Las honeypots se pueden clasificar en base a los siguientes aspectos: nivel de interacción, tipo de honeypot e implementación de la honeypot.

La clasificación por nivel de interacción con el usuario tiene tres posibles diferenciaciones:

- **Baja interacción:** Las honeypots de baja interacción son aquellas en las que la honeypot apenas da interacción con el usuario. Básicamente trata de ofrecer servicios emulados como SSH, FTP, Telnet, HTTP, etc. Los atacantes lanzan sus ataques contra estos servicios y la honeypot recoge toda la información posible de sus acciones.
- **Media interacción:** Las honeypot de interacción media, al igual que las de baja interacción, tienen un comportamiento limitado de cara al atacante. Sin embargo, en estas sí que se llega a dar algo más de funcionalidad a los servicios para que los atacantes encuentren un entorno más realista.
- **Alta interacción:** Las de alta interacción son las más complicadas de diseñar e implementar. Su nivel de sofisticación es alto, ya que todo lo que hay en ella no es simulado y permite interacción plena por parte del atacante. Son excelentes para recoger mucha información e información completa. Como desventaja, estas honeypots aportan riesgos a la infraestructura real si no se controlan.

Además de los niveles de interacción, existen dos tipos de honeypot clasificadas según el tipo de uso:

- **Honeypot de investigación:** Utilizadas por instituciones educativas y de investigación. La información recogida de los ataques viene motivada ante la posibilidad de descubrir nuevos ataques o las motivaciones que llevan al atacante a cometer estos actos.

- **Honeypot de producción:** Se utilizan en medianas y grandes empresas y tienen como objetivo desviar los ataques contra esta. Además, al igual que en las de investigación, también buscan recabar información aunque se centran más en averiguar la motivación que lleva al atacante a fijarles como objetivo.

Para terminar, las honeypot también pueden diferenciarse en la forma de implementación:

- **Honeypot virtual:** Como su propio nombre indica, es una honeypot virtualizada mediante software de virtualización y alojada dentro de un host físico. Esta honeypot está perfectamente integrada dentro de la propia red de la infraestructura contando con su propia configuración de red. Estas honeypot son sencillas de implementar y más económicas que las físicas.
- **Honeypot física:** A diferencia de la honeypot virtual, esta honeypot cuenta con un equipo específico para su cometido y al igual que la virtual, se encuentra perfectamente integrada dentro de la red. La ventaja que aporta esta honeypot es que es lo más realista posible, aunque sin embargo el coste de mantenimiento es más elevado que una honeypot virtual.

Otro elemento destacable y que es crucial para el desarrollo del trabajo son las honeynet. Las honeynet son redes de honeypot que son capaces de emular una infraestructura completa para dotar de más realismo y realizar una recogida de información lo más completa posible. A modo de ejemplo gráfico, la Figura 2.3 muestra un esquema de honeynet.

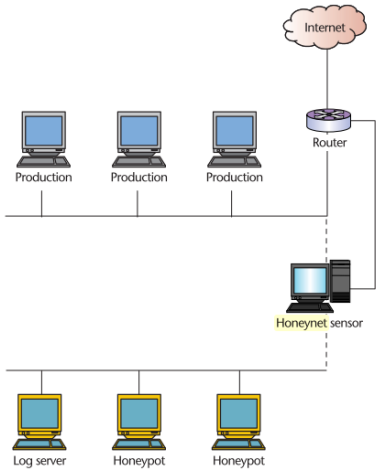

Figura 2.3: Esquema Honeynet. Fuente: [22].

En este trabajo se va a realizar una honeynet con finalidad puramente investigadora. La implementación es totalmente virtualizada y se desplegará en un equipo local, haciendo que sea fácilmente integrable dentro de una red industrial IoT. Para su desarrollo, la combinación de honeypots de baja y media-alta interacción permiten emular servicios para dotar de realismo la red, recogiendo registros de interacciones y también muestras maliciosas.

2.3. Malware

Se define como malware a todo software diseñado para realizar actividades maliciosas en el sistema objetivo. Este software puede ser un ejecutable, un script, código o cualquier otro tipo de software. Las actividades maliciosas que buscan los atacantes abarcan desde el robo de información confidencial, espionaje de un sistema infectado hasta incluso tomar el control del mismo, y todas comúnmente con el mismo fin, obtener beneficio económico.

El malware puede ser clasificado en base a sus características, algunas de las más destacables son sus mecanismos de propagación y la funcionalidad de la que está dotado.

Respecto a la propia funcionalidad del malware, se puede clasificar en diferentes tipos, como por ejemplo: [15]:

- **Virus y gusanos:** Estos tipo de malware tienen la capacidad de replicarse y esparcirse a otros equipos. En el caso de los virus es necesaria interacción con el usuario, a diferencia de los gusanos.
- **Troyanos:** Malware que se instala en la máquina víctima y puede realizar acciones maliciosas como robo de información y envío de ficheros al servidor del atacante, entre otros.
- **Backdoor:** Malware similar al troyano, aunque consigue acceso al atacante para ejecutar comandos en el sistema.
- **Botnet:** Una botnet es un grupo de máquinas infectadas con el mismo malware y que obedecen a las instrucciones que se les envía desde un servidor *Command and Control* controlado por el atacante.
- **Ransomware:** Este malware cifra los archivos del sistema infectado y a cambio pide una recompensa para recuperar la información.

Todos estos tipos de malware explicados son los que se pueden apreciar en la Figura 1.2 descrita en el capítulo 1 y son los más destacables dentro del entorno IIoT.

El análisis de malware es el estudio del comportamiento del malware con la idea de obtener información acerca de su funcionamiento y así buscar soluciones para detectarlo y eliminarlo. El análisis de malware se clasifica en dos tipos:

- **Análisis estático:** Es el proceso de análisis de ficheros de malware sin su ejecución. En esta fase se siguen procedimientos para determinar el tipo de archivo, obtener la huella digital del malware, extraer cadenas de texto que contengan información relevante y hacer un análisis del flujo de control.
- **Análisis dinámico:** Este análisis se realiza ejecutando el malware a analizar. Ejecutada la muestra maliciosa, se monitorizan los procesos para comprobar la interacción con el sistema operativo, se analiza el registro del sistema y el sistema de ficheros. También se monitorizan las redes para comprobar si existe algún tipo de interacción con el exterior.

Capítulo 3

Estado del arte

En este capítulo se describen dos apartados, por un lado el estado actual de las honeypot tanto en dispositivos IoT como en ICS. Por otro lado, el estado actual del malware en IoT. El primer apartado comenzará con una revisión de la literatura mencionando aquellas honeypot más relevantes en entornos industriales, siguiendo con las del campo del IoT. El segundo apartado se basa en un análisis del malware actual en dispositivos IoT.

3.1. Honeypots

3.1.1. Honeypots industriales

Comenzando con la parte industrial, en [19] se presenta Honeyd, un framework para honeypots virtuales que simula sistemas a nivel de red. El objetivo principal es responder a los paquetes con destino a las direcciones IP que conforman la honeypot. Este framework dispone de un proceso residente que recibe paquetes de la red que son procesados por un gestor de paquetes interno que solamente trabaja con tres protocolos: ICMP, TCP y UDP; el resto de paquetes son descartados.

En [11] se realiza un análisis sobre Conpot, una honeypot orientada a sistemas SCADA. Se trata de una honeypot de baja interacción e implementación muy sencilla utilizando un sistema de registro que recoge datos como dirección de origen, tipo de petición y según el protocolo, qué recurso se ha solicitado.

3.1.2. Honeypots en IoT

Siguiendo con el análisis de honeypots para IoT, en [17] se analiza una honeypot llamada IoTPOT centrada en ataques contra el servicio Telnet. El incremento disparado de ataques llevados a cabo contra dispositivos IoT durante el año 2014 sirve de motivación para este sistema.

En [14] se estudia una honeypot un tanto sofisticada motivada por la heterogeneidad de los dispositivos IoT. En concreto, se describe la forma de buscar una honeypot de interacción inteligente de tal forma que sea capaz de aprender el funcionamiento de varios tipos de dispositivos mediante técnicas de aprendizaje máquina.

Para concluir, tal y como se ha visto en el análisis de las honeypots en esta sección, están todas centradas en un campo, bien sea emulando sistemas o dispositivos IoT. El aporte que se quiere dar con este trabajo es crear una red que combine algunas de las honeypots ya existentes de código abierto y con diferentes funcionalidades. De esta forma, se puede crear un sistema más amplio que permita recoger información lo más completa posible y de diferentes tipos.

3.2. Malware

En lo que a la parte de malware respecta, en [2] se describen una serie de malwares que afectan a los dispositivos IoT destacando tres familias de Botnets: Gagfyt, Kaiten y Mirai. La Tabla 3.1 muestra las características más importantes de estas tres familias.

Tabla 3.1: Resumen de las tres familias de malware IoT más características

Mirai	Kaiten	Gagfyt
▪ Se construye con el propósito de ser vendida como herramienta para realizar DDoS.	▪ Sus servidores de control están basados en el protocolo IRC.	▪ Sus servidores de control están basados en el protocolo TCP.
▪ Código abierto que puede dar lugar a creación de variantes más peligrosas.	▪ Soporte para múltiples arquitecturas.	▪ Soporte para múltiples arquitecturas.
▪ Está diseñada de tal forma que es capaz de eliminar otras aplicaciones competidoras.	▪ Las variantes más recientes implementan la posiblidad de eliminar otras botnets.	▪ Las versiones más recientes tienen la posibilidad de desinstalar otras botnets.

En [1] se analiza en más detalle Mirai, malware destacable contra IoT cuyo objetivo es infectar el mayor número de dispositivos posible para después llevar a cabo

un ataque DDoS contra otros objetivos. Para infectar, utiliza el protocolo TCP/IP mediante Telnet y utilizando un diccionario que almacena en un servidor. Una vez se propaga, elimina todos los procesos que se ejecuten en los puertos TCP 22 y 23.

En [2] se analiza Brickerbot, que realiza un ataque conocido como PDoS (*Permanent Denial of Service*). El ataque utiliza busybox, un conjunto de utilidades Unix en un único ejecutable, con el que bloquea e inutiliza un dispositivo. Para comprometer la máquina utiliza Telnet por el puerto 23. Otro de los puertos que también utiliza es el 22 con el servicio SSH.

Como se puede ver, la mayoría de malware se aprovecha de los servicios SSH y Telnet para llevar a cabo procesos de infección. Esto se debe a que estos dispositivos en numerosas ocasiones vienen con escasa protección de fábrica y no se aplican mecanismos de seguridad, lo que los hace vulnerables contra estos ataques.

Para concluir el capítulo, cabe destacar que a pesar de que en la honeynet se emulen diferentes servicios, la revisión de literatura anterior deja claro que se debe centrar la atención en los servicios SSH y Telnet. Ambos son los principales puntos de entrada de malware y por tanto elementos clave en la recogida de muestras. Por ello, el principal foco de interacción con la honeynet desarrollada en este trabajo se da en estos servicios, concretamente en SSH más que por Telnet.

Capítulo 4

Análisis, diseño e implementación

El presente capítulo se divide en tres fases que abordan los aspectos técnicos necesarios para llevar a cabo el desarrollo del trabajo. En la fase de análisis se realiza un estudio sobre los elementos necesarios más adecuados para lograr el objetivo. Después, en la fase de diseño se da forma a toda la infraestructura de la honeynet a desarrollar. Por último, la fase de implementación describe el proceso seguido para llevar a cabo la implementación de todo el sistema, describiendo las tecnologías utilizadas y el repositorio de software creado.

4.1. Análisis

En esta sección se busca el objetivo a conseguir con la creación de la honeynet y a partir de este se analizan los aspectos necesarios para llevar a cabo su desarrollo.

4.1.1. Objetivo del sistema

Antes de comenzar el análisis, es importante determinar el tipo de honeynet que se quiere implementar. Como se ha comentado en la Sección 2.2, el objetivo de este trabajo es construir un sistema de honeypots de investigación que permita recoger información orientada al análisis de malware en IIoT.

Para llevar a cabo el desarrollo del sistema, se deben tener en cuenta algunos requisitos que debe cumplir para lograr su cometido tanto engañando a los atacantes como en la recogida de información.

Los más importantes a tener en cuenta en el desarrollo son:

- **Realismo:** Dotar de realismo a la honeynet reduce la posibilidad de que ésta sea detectada. Para ello, se debe recrear una arquitectura similar a las que se pueden encontrar en entornos industriales.
- **Recogida de información útil:** Definir los datos que se deben recoger es importante ya que esta información servirá de estudio para identificar nuevas amenazas. Esta información se puede clasificar en dos categorías: *información de movimientos a nivel de red*, se trata de controlar qué movimientos se realizan y en qué punto de la red se realizan, y la otra *información de interacción con honeypot*. Esta última engloba los intentos de autenticación y descargas del atacante, entre otros.

4.1.2. Infraestructura

Ya con el objetivo del sistema en mente, se continúa analizando una infraestructura industrial IoT de cara a determinar qué honeypots de las existentes y de código abierto pueden ayudar a recrear un sistema industrial IoT lo más realista posible. En la Figura 4.1 se muestra una arquitectura IIoT de alto nivel de la que se extraen las ideas de organización y de elementos a utilizar. Por tanto, los elementos necesarios son los que corresponden con la red industrial: *SCADA Local*, *IIoT Gateway* y *PLC*.

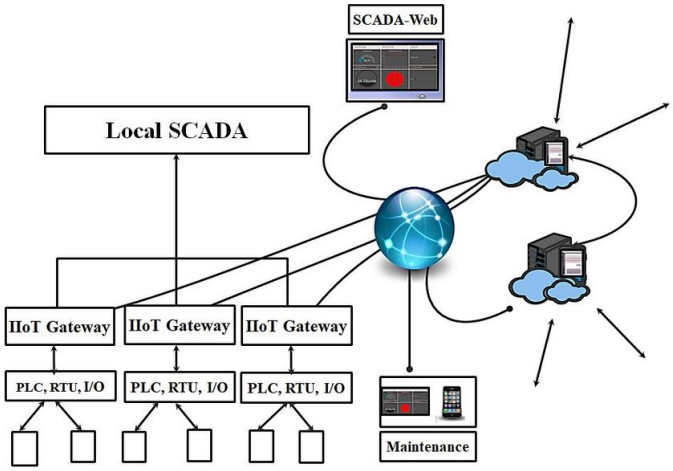

Figura 4.1: Arquitectura ICS. Fuente: [9].

Tanto los elementos de control de los dispositivos como los SCADA están conectados a un *IIoT Gateway* para que transmita la información al exterior. Por tanto, se necesita una red industrial que comunique todos los elementos con el gateway de forma que se cumpla con esta topología. Otro aspecto a tener en cuenta es que el gateway debe comunicar también con el exterior, dado que es la puerta de entrada y salida.

Como primer punto del análisis, conviene comenzar con el gateway por ser un elemento de gran importancia, ya que es la puerta de entrada al sistema. Este elemento debe mostrar un aspecto realista para que sea objetivo de ataques, además su conexión con el exterior hace necesario que sea interactivo con los posibles atacantes. Para ello, se toma como ejemplo la Figura 4.2.

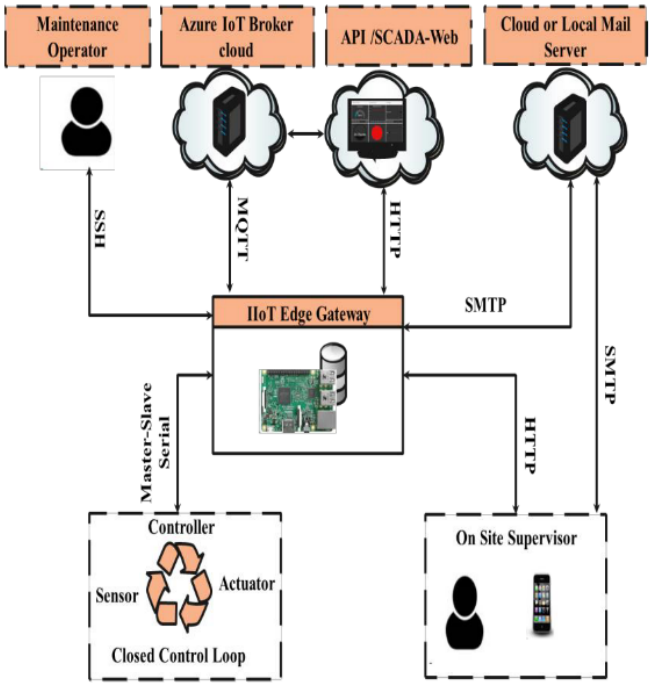

Figura 4.2: Esquema IIoT Gateway. Fuente: [9].

Esta figura muestra el esquema propuesto en [9] para crear un entorno de pruebas que recoja muestras de malware y del que se extraen los protocolos a utilizar en el gateway, concretamente:

- **SSH:** Para operaciones de mantenimiento.
- **MQTT:** Protocolo para comunicar con el cloud.
- **HTTP:** Servidor web para SCADA.
- **SMTP:** Servidor de correo.

Finalmente, el gateway deberá ser un elemento independiente del resto, de esta forma se asegura dotarle de interacción total con los atacantes.

Continuando con el análisis sobre la red industrial, las honeypots industriales presentadas en el Capítulo 2 están capacitadas para llevar a cabo una implementación de este tipo.

Para tomar una decisión sobre qué honeypot de las de código fuente disponibles se ha de considerar en este trabajo se han comparado entre ellas, según se muestra en la Tabla 4.1.

Tabla 4.1: Comparativa de las honeypots industriales

Honeypot	Descripción	Ventajas	Desventajas
Honeyd	Con un sólo host puede recrear topologías de red completas y asignando diferentes IP a los dispositivos emulados. Se utiliza como base para otras honeypots	▪ Totalmente personalizable ▪ Escalabilidad ▪ No depende de hardware	▪ Antiguo ▪ Repositorio sin actualizar
Conpot	Simula sistemas industriales incluyendo PLCs y otros controladores. También es capaz de simular protocolos como Modbus o SNMP.	▪ Facilidad de despliegue ▪ Variedad de protocolos ▪ Buen sistema de registro ▪ No depende de hardware	▪ Predefinido

La honeypot que se ha decidido utilizar para el desarrollo del trabajo ha sido Honeyd. La elección viene motivada por la posiblidad que ofrece de personalizar absolutamente la topología de red. Esta posibilidad, junto con la de implementar una red con diversos dispositivos en un único host, abre la posibilidad de crear contenedores con redes industriales.

Para terminar este apartado, se analiza el elemento más importante del sistema, que es la simulación de un dispositivo IoT vulnerable. Existen diferentes honeypots que emulan dispositivos IoT y recogen información sobre los ataques como se han explicado en la Sección 3.1. La Tabla 4.2 muestra la comparativa realizada entre ellas.

De las honeypots expuestas en la tabla, *IoTPot* se descarta puesto que no tiene repositorio por lo que resulta imposible su implementación. Respecto a *Telnet Iot Honeypot*, es una honeypot muy interesante para el objetivo del sistema, sin embargo

Tabla 4.2: Comparativa de las honeypots en IoT

Honeypot	Descripción	Ventajas	Desventajas
IoTPot	Honeypot que emula arquitecturas con QEMU para MIPS,ARM, PPC. Ofrece diferentes tácticas de login.	▪ No depende de hardware	▪ No hay repositorio
Telnet IoT Honeypot	Honeypot que simula un dispositivo IoT en el que expone el puerto de Telnet para recibir ataques. Recoge malware y lo analiza, mostrándolo en un *frontend*.	▪ Recoge malware y lo analiza ▪ No depende de hardware	▪ Solo se centra en Telnet
Cowrie	Honeypot que ofrece un sistema de login flexible y recoge información de cada uno de estos. Puede simular diferentes sistemas de ficheros y directorios. Registros de todas las interacciones de los atacantes, guardando copias de las descargas que se hagan con herramientas para capturar contenido de servidores web, como por ejemplo wget o curl, entre otros.	▪ Actualizada ▪ Buena recogida de info y de ficheros ▪ No depende de hardware	▪ No es específica de un IoT

sólo se enfoca en Telnet cerrando posibles entradas alternativas como puede ser SSH. Se decide por tanto utilizar *Cowrie*, ya que se pueden obtener los mismos resultados que con *Telnet Iot Honeypot* incluso por SSH. Además, se puede recoger la información de login y las descargas realizadas por los atacantes, algo de especial interés para analizar malware de la familia de botnets.

4.1.3. Gestión de información

Para la gestión de información se ha usado el sistema ELK (Elasticsearch, Logstash, Kibana) para simplificar y facilitar el análisis de la información obtenida en el sistema. La Figura 4.3 muestra la arquitectura de componentes del ELK:

Figura 4.3: Esquema ELK Stack. Fuente: [13].

El flujo que se necesita en este sistema consiste en cuatro pasos: (1) Recopilación de logs de cada punto de interés; (2) Recogida de logs, transformación y envío a Elasticsearch; (3) Indexado de los logs recibidos por Logstash y envío a Kibana; y (4) Visualización y monitorización de logs en Kibana.

4.2. Diseño

En esta sección se lleva a cabo el diseño del sistema tomando como referencia el análisis realizado anteriormente. Se comienza explicando la arquitectura general del sistema y posteriormente se desarrolla cada uno de los elementos que lo conforman.

4.2.1. Arquitectura del sistema

Como se muestra en la Figura 4.4, la arquitectura general del sistema se divide en dos partes generales y claramente diferenciadas. Una de ellas es el host donde va a correr todo el sistema y la otra es la parte de contenerización.

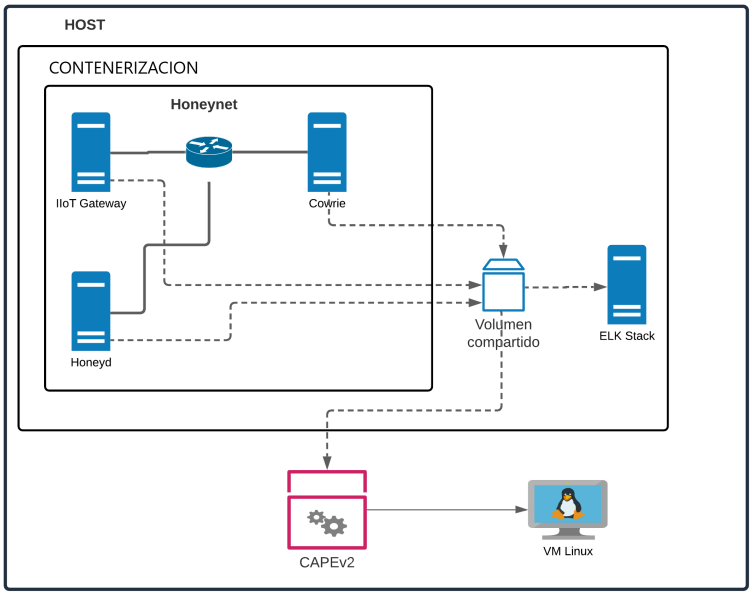

Figura 4.4: Esquema general del sistema. Fuente propia

4.2.2. Host

Es la base del sistema, se dispondrá en una máquina que soporte su funcionamiento y que use una distribución GNU/Linux. Se mantendrá aislado de la parte de contenerización para evitar intrusiones, a excepción de la conexión con los contenedores mediante SSH en caso de necesitar labores de mantenimiento, y el punto de unión con un volumen compartido donde se recogerán las muestras de descargas realizadas por los atacantes dentro del sistema. Para el análisis de dichas muestras, se mantiene en segundo plano un demonio ejecutando el sandbox CAPEv2 de modo que cada cierto tiempo se recogen las muestras del volumen compartido y se analizan. Cabe destacar que se deberá disponer de un software de virtualización para que CAPEv2 pueda ejecutar las muestras maliciosas.

4.2.3. Contenerización

La contenerización se divide en cuatro contenedores diferentes, tres de ellos conectados entre sí y que conforman la honeynet y por otro lado está el contenedor creado para el ELK stack. La información entre la honeynet y el ELK stack se comparte mediante el uso de volúmenes de datos generados entre contenedores.

La Figura 4.5 muestra la arquitectura de la honeynet. En la imagen se pueden ver las tres partes de las que consta. Cada una de estas partes consiste en un contenedor diferente, para facilitar la escalabilidad y la gestión de la honeynet.

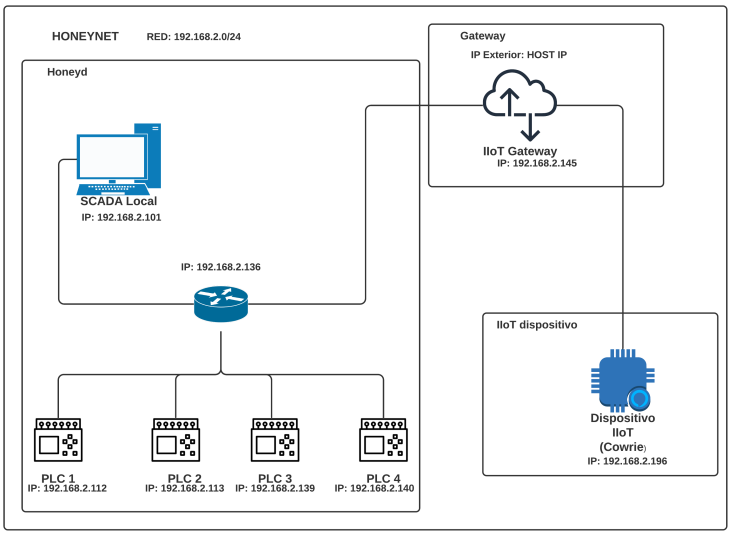

Figura 4.5: Arquitectura honeynet. Fuente propia

A nivel de comunicación, la arquitectura se ha encapsulado dentro de una misma red, de tal forma que cuando el atacante consiga acceso al gateway y haga un mapeo, tenga sensación de haber accedido dentro de una red industrial. La IP asignada a la red es 192.168.2.0/24.

El gateway es un elemento que hace de puerta de entrada y de salida a la red. El contenedor dispondrá de dos interfaces de red, la del host y la de red industrial que comunica los contenedores. Hay que destacar que los puertos del host han de ser reconfigurados para dejar expuestos los puertos del contenedor, de esta forma se queda expuesto al exterior el gateway y no el host local. Esto es necesario en caso de que el host local deba mantener estos servicios abiertos.

El contenedor en el que se ejecuta Honeyd representa una red industrial y es la parte ICS del sistema. Consta de los elementos que se encuentran en la Tabla 4.3.

El otro contenedor dispone de Cowrie y representa un dispositivo IoT. En este caso se plantea un problema, ya que al utilizarse el puerto 22 para SSH por Cowrie se debe disponer de un puerto alternativo para SSH en caso de labores de mantenimiento del contenedor. Para ello, se modifica el fichero de configuración de SSH en el contenedor y se cambia por el puerto 45022, de esta forma los accesos por el puerto 22 se redirigen a Cowrie quedando ocultos los que van por el puerto 45022.

Cada uno de los contenedores que conforman la honeynet es un punto de recogida de información que será enviado al volumen compartido. A la vez que se envía

información al volumen, el ELK se encarga de recogerlos y procesarlos para facilitar la gestión de información.

Tabla 4.3: Configuración de elementos en honeyd

	Personalidad	IP	Puertos abiertos
SCADA	Microsoft Windows 7 Professional	192.168.2.101	▪ 22 ▪ 23 ▪ 80 ▪ 502 ▪ 44818
Router	Cisco Catalyst 2960 or 3600 switch (IOS 12.3)	192.168.2.136	▪ 22 ▪ 23
PLC1	.Allen-Bradley 1756 ControlLogix programmable automation controller	192.168.2.112	▪ 80 ▪ 44818
PLC2	Allen-Bradley 1756 ControlLogix programmable automation controller	192.168.2.113	▪ 80 ▪ 44818
PLC3	Siemens Simatic 300 programmable logic controller	192.168.2.139	▪ 80 ▪ 502
PLC4	Siemens Simatic 300 programmable logic controller	192.168.2.140	▪ 80 ▪ 502

4.3. Tecnologías

Para la máquina host se ha utilizado un sistema operativo Debian 10 "Buster" [5] sobre la que se han instalado las siguientes herramientas:

- Docker [6], que es un proyecto de código abierto que permite crear y desplegar aplicaciones dentro de contenedores. Esta herramienta se ha utilizado para la contenerización.
- Docker-compose [16], que es una herramienta que simplifica el uso de Docker gracias a la posibilidad de configurar los contenedores mediante ficheros YAML, un formato de serialización de datos. Con esta herramienta se han gestionado los contenedores simplificanado notoriamente la gestión.
- CAPEv2 [4], que es un sandbox utilizado para análisis de malware y es la aplicación que se encuentra ejecutada en segundo plano esperando el envío de muestras de código dañino para ser analizadas.

4.4. Repositorio del sistema desarrollado

La herramienta se encuentra alojada en GitHub, donde se puede encontrar la documentación respecto a su uso. El código se ha liberado bajo licencia GNU/GPL v3 y está disponible en el siguiente enlace:

`https://github.com/ibaitxo/iiot-honeynet`.

Capítulo 5

Pruebas de evaluación

En este capítulo se realiza una evaluación del sistema desarrollado. Se dividirá en dos partes, por un lado se explicarán las pruebas realizadas a la honeynet y se evaluará su funcionamiento respecto a los objetivos de monitorización de eventos y recogida de posibles muestras maliciosas. Por otro lado, se harán pruebas de análisis de malware sobre dichas muestras recogidas en la honeynet y se evaluará si el sistema funciona adecuadamente.

5.1. Escenarios de evaluación

Las pruebas se han llevado a cabo sobre un host local funcionando sobre un sistema operativo Debian 11 "Bullseye" sobre el que correrá todo el sistema (honeynet, ELK Stack y el CAPEv2). Para emular los ataques, se ha utilizado una máquina virtual utilizando VirtualBox como software de virtualización y Kali Linux como sistema operativo para los ataques.

Para llevar a cabo el proceso, se han utilizado las herramientas del sistema operativo Kali Linux. Al ser un sistema operativo enfocado a la ciberseguridad, proporciona herramientas que permitirán realizar análisis de redes e inicios de sesión por fuerza bruta, entre otras cosas.

Los casos de prueba que se plantean irán acordes al elemento a analizar en cuestión. Para los elementos de la honeynet, las pruebas se centrarán en escaneos de redes, intentos de inicio de sesión utilizando diccionarios, y descargas de elementos maliciosos. Las pruebas que se realizan sobre el sistema de análisis de malware irán enfocadas sobre su funcionamiento: envío de la muestra al sandbox, almacenado de las muestras y resultados del análisis.

5.2. Pruebas honeynet

Las pruebas se han realizado simulando el entorno desde local, presuponiéndose que la dirección del gateway ya ha sido descubierta por el atacante.

5.2.1. IIoT Gateway

La primera prueba va diriga contra el IIoT Gateway. Desde la máquina atacante se ha realizado un escaneo para ver qué puertos están abiertos en el dispositivo. Para ello, se utiliza la herramienta *nmap*, una herramienta de código abierto que permite realizar escaneos de puertos y descubrimiento de equipos. Con ella se rastrearán los puertos abiertos en el dispositivo para encontrar una forma de entrar en él. Como resultado del escaneo, el puerto 23 correspondiente a SSH se muestra abierto, por lo que se lanza un ataque por diccionario para comprobar que el sistema de registros recoge todos estos intentos. Como resultado, se obtiene el fichero auth.log, un fichero generado por SSH que proporciona información de todos los intentos de autenticación que se han dado en el servicio. Este fichero es el que se envía al ELK para ser indexado y mostrar en Kibana la información. Las muestras de los intentos de conexión se encuentran en el Anexo C.

5.2.2. Honeyd - ICS

Cuando un atacante ha conseguido autenticación y acceso al gateway, éste ya se encuentra en la red interna. Ahora puede pivotar por la red y escanearla en busca de más objetivos. Todo esto será recogido mediante *Honeyd*. Para este apartado, se plantean dos pruebas diferentes: (1) Escaneo de la red con Nmap y (2) Análisis de dispositivo.

Escaneo de la red

Antes de comenzar conviene detallar la estructura que siguen los logs que genera *Honeyd* y que se puede apreciar en la Tabla 5.1.

Tabla 5.1: Formato de log generado por honeyd

Timestamp	Protocolo	Flag	Origen IP Puerto	Destino IP Puerto	Info	Comentario
2021-08-31-17:39:47.8303	tcp(6)	S	192.168.2.145 33854	192.168.2.140 5353	88 PA	

Tal y como se muestra en la Figura 5.1, la honeypot ha comenzado a escribir en el fichero de log. La información que muestra no es relevante pero indica que se ha iniciado correctamente.

```
root@45958fafac2c:/home/admin/Honeyd/logs# cat honeyd.log
2021-08-30-21:07:47.0738 honeyd log started ------
2021-08-30-21:07:50.9090 udp(17) - 192.168.2.254 5353 224.0.0.251 5353: 73
2021-08-30-21:07:50.9092 udp(17) - 192.168.2.254 5353 192.168.2.112 5353: 73
2021-08-30-21:07:50.9092 udp(17) - 192.168.2.254 5353 192.168.2.113 5353: 73
2021-08-30-21:07:50.9092 udp(17) - 192.168.2.254 5353 192.168.2.139 5353: 73
2021-08-30-21:07:50.9092 udp(17) - 192.168.2.254 5353 192.168.2.140 5353: 73
2021-08-30-21:08:22.8770 udp(17) - 192.168.2.254 5353 224.0.0.251 5353: 73
2021-08-30-21:08:22.8771 udp(17) - 192.168.2.254 5353 192.168.2.112 5353: 73
2021-08-30-21:08:22.8771 udp(17) - 192.168.2.254 5353 192.168.2.113 5353: 73
2021-08-30-21:08:22.8771 udp(17) - 192.168.2.254 5353 192.168.2.139 5353: 73
2021-08-30-21:08:22.8771 udp(17) - 192.168.2.254 5353 192.168.2.140 5353: 73
```

Figura 5.1: Inicio de escritura de los ficheros logs de Honeyd. Fuente propia

Para escanear la red, se utiliza la herramienta *nmap* para que busque entre un intervalo de direcciones IP. A medida que el escaneo de la red avanza, *Honeyd* recoge información de todos estos movimientos realizados y los escribe el fichero honeyd.log tal y como se ve en la Figura 5.2.

```
2021-08-30-21:15:26.0929 tcp(6) - 192.168.2.145 34031 192.168.2.113 6788: 44 S
2021-08-30-21:15:26.2010 tcp(6) - 192.168.2.145 34032 192.168.2.139 32776: 44 S
2021-08-30-21:15:26.2010 tcp(6) - 192.168.2.145 34032 192.168.2.140 3269: 44 S
2021-08-30-21:15:26.2011 tcp(6) - 192.168.2.145 34032 192.168.2.112 464: 44 S
2021-08-30-21:15:26.2370 tcp(6) - 192.168.2.145 34030 192.168.2.136 50001: 44 S
2021-08-30-21:15:26.3450 tcp(6) - 192.168.2.145 34033 192.168.2.139 32776: 44 S
2021-08-30-21:15:26.3810 tcp(6) S 192.168.2.145 34157 192.168.2.112 80
2021-08-30-21:15:26.3812 tcp(6) - 192.168.2.145 34031 192.168.2.136 50001: 44 S
2021-08-30-21:15:26.4170 tcp(6) E 192.168.2.145 34157 192.168.2.112 80: 0 0
```

Figura 5.2: Captura de logs Nmap. Fuente propia

Siguiendo con las pruebas de escaneo, se ha realizado el análisis al host que representa el conmutador "Cisco Catalyst" para comprobar si se identifica como tal. El resultado obtenido ha sido satisfactorio, se consigue mostrar el dispositivo con la identidad correspondiente que se le ha dado en la configuración. En la Figura 5.3 se aprecia el resultado obtenido.

Análisis de dispositivo

```
Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-02 05:53 UTC
Nmap scan report for 192.168.2.136
Host is up (0.015s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE    VERSION
22/tcp open  tcpwrapped
23/tcp open  tcpwrapped
MAC Address: 08:4F:A9:B1:5D:81 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: switch
Running: Cisco IOS 12.X
OS CPE: cpe:/h:cisco:catalyst_2960 cpe:/h:cisco:catalyst_3600 cpe:/o:cisco:ios:12.3
OS details: Cisco Catalyst 2960 or 3600 switch (IOS 12.3)
Network Distance: 1 hop
```

Figura 5.3: Resultado del escaner al host. Fuente propia

5.2.3. Cowrie - IoT Device

Este elemento es clave en el sistema, dado que en él se recogen las muestras de malware que un atacante trate de utilizar y el proceso seguido para llevarlo a cabo. Para comprobar que esto funciona se realizarán dos pruebas: (1) Registro de inicios de sesión y (2) Recogida de muestras.

Registro de inicios de sesión

La primera prueba a realizar consistirá en realizar un intento de autenticación con diccionario para recoger las credenciales que han sido utilizadas. En esta prueba se conoce la dirección. Sin embargo, en un ataque real se harían búsquedas de puertos abiertos hasta dar con uno. La Figura 5.4 muestra los puertos abiertos del dispositivo.

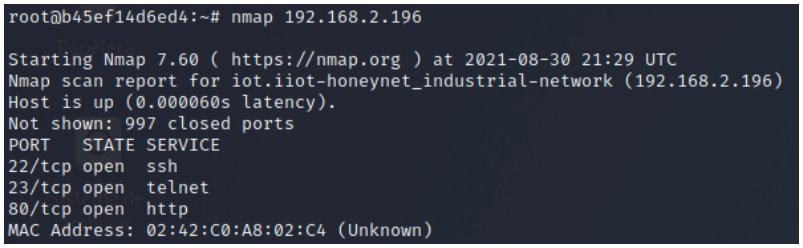

Figura 5.4: Búsqueda de puertos abiertos en IoT. Fuente propia

Los diferentes intentos de autenticación se realizan con diccionarios hasta conseguir el acceso. Para esta acción, se utilizará una herramienta para comprobar diferentes credenciales (en este caso, se utiliza hydra [8]). Para realizar la prueba, se ejecuta la herramienta con un diccionario de contraseñas. Si todo es correcto, cada uno de los intentos de autenticación quedarán reflejados en los logs de Cowrie y se mostrarán en Kibana. Estas muestras se encuentran en el Anexo C.

Recogida de muestras

La recogida de muestras se hace automáticamente por parte de *Cowrie*. Todo elemento que sea descargado mediante *wget* o similares, o incluso haciendo una transferencia con *scp* , se almacena en el directorio *"/var/lib/cowrie/downlaods/binario"* de la honeypot.

Las pruebas planteadas en este caso son dos: descarga de binario mediante *wget* y transferencia de binario a través de *scp*.

Descarga de binario

Una vez se accede al sistema, se procede a realizar una petición *web* sobre *http://123.10.16.81:33235/Mozi.m*, dirección obtenida de `https://urlhaus.abuse.ch/` para obtener el malware. Una vez completada la descarga, en el directorio de almacenamiento de muestras en *Cowrie* se ha creado una copia del binario con su hash SHA-256 (*4293c1d8574dc87c58360d6bac3daa182f64f7785c9d41da5e0741d2b1817fc7*).

Transferencia de binario mediante SCP

Una vez que se conocen las credenciales de la máquina objetivo, el atacante puede realizar una transferencia del binario utilizando SCP. La operación se realizó con éxito, aunque al entrar de nuevo en la honeypot, el fichero no existe. Esto es debido a que por cada sesión la honeypot elimina todos los archivos que no son necesarios. No obstante, sí que ha sido almacenada una copia en el directorio de muestras con su hash SHA-256 (*2524b2e02bd5a75336389b7b4d7b6660a24879642cbed0ab60681423e795cd49*).

5.2.4. Problemas encontrados

En la prueba de descarga de binario, como primera opción se realiza la descarga con *wget*. Sin embargo, la descarga no se lleva a cabo debido a que no se consigue conectar con el servidor. Como alternativa, se procede a utilizar la herramienta *curl*, realizando la operación satisfactoriamente.

5.3. Pruebas de sistema de análisis

En esta sección se explica el funcionamiento del sistema de análisis. Al ser un sistema que funciona en segundo plano, la prueba consiste en comprobar que se realizan correctamente todas las operaciones.

5.3.1. Proceso

Antes de comenzar, se debe asegurar que se encuentran funcionando en segundo plano tanto CAPEv2 como la herramienta creada para automatizar el proceso. También se ha debido crear la interfaz de red que conecta con la máquina virtual "vboxnet0". Estos procesos se ejecutan mediante cron, el administrador de tareas de GNU/Linux y que tiene programados los inicios de tareas cuando se enciende el equipo.

Con todo en funcionamiento, una tarea de cron se activa cada cinco minutos. Al activarse, se comprueba que realiza una copia del primer elemento del directorio donde se guardan las muestras de la honeynet, localizado en "~/honeynet/muestras/downloads". Después, se mueve a otro directorio de almacenamiento llamado "BibliotecaMuestras" dentro del directorio *honeynet*, borrándose el de origen. La copia del fichero original se envía a CAPEv2 para que realice el análisis, que una vez terminado, guarda en el directorio "~/CAPEv2/storage/analysis/id tarea" un reporte con los datos obtenidos. Para demostrar su funcionamiento, se ha incluido un ejemplo del informe obtenido en el Anexo C.

5.3.2. Problemas encontrados

Debido a una actualización de la distribución durante las pruebas, se perdió la funcionalidad de cron, por lo que los procesos y la configuración de red de la máquina virtual estaban inactivos. No obstante, se trataba de un problema menor que se ha solucionado, pero es un aspecto a tener en cuenta durante el funcionamiento.

Respecto al análisis de malware, algunos binarios disponían de una arquitectura ARM de la que no se podía realizar análisis al no disponer de una máquina virtual. Como solución se ha utilizado una máquina virtual de Raspberry Pi que emula un dispositivo con arquitectura ARM.

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo se reflejan las conclusiones del trabajo desarrollado y se habla también acerca de posibles mejoras en la herramienta como continuación del desarrollo.

La implementación del IoT dentro del mundo industrial ha supuesto una mejora de los procesos, pero a su vez han proliferado los ataques dirigidos contra estos. Este trabajo se ha centrado en los ataques de tipo malware, los cuales durante el año 2020 representaron un tercio del total de amenazas contra IoT. El objetivo final de este trabajo ha sido desarrollar una honeynet que se integrara dentro de un entorno IoT Industrial y que colaborase en obtener información sobre estos ataques malware.

El desarrollo de esta honeynet se ha realizado para conseguir un sistema lo más realista posible para engañar a los atacantes. En el caso de este trabajo, se han seguido varios objetivos para cumplir con un buen desarrollo, dando lugar a resultados satisfactorios y que cumplen con el cometido.

Como trabajo futuro, en primer lugar se deberían realizar pruebas en un entorno real para comprobar qué comportamiento tienen los atacantes sobre ella y si es necesario remodelar algo. Respecto al sistema de análisis, una posible solución para los problemas de arquitectura a la hora de ejecutar binarios consistiría en disponer de un sistema de clasificación por el tipo de arquitectura a medida que se van recogiendo muestras, de modo que primero las preprocesara para clasificarlas y posteriormente enviarlas a la máquina virtual más adecuada. Además, disponer de máquinas virtuales configuradas con diferentes arquitecturas está soportado por CAPEv2, con lo que se podría aumentar el rango de muestras analizadas de manera muy sencilla.

Bibliografía

[1] Manos Antonakakis y col. «Understanding the Mirai Botnet». En: *26th USENIX Security Symposium (USENIX Security 17) Security 17)*. Vancouver, BC: USENIX Association, 2017, págs. 1093-1110. ISBN: 978-1-931971-40-9.

[2] Michael Austin. «IoT Malicious Traffic Classification Using Machine Learning». Tesis doct. West Virginia, 2021.

[3] Sravani Bhattacharjee. *Practical Industrial Internet of Things Security: A practitioner's guide to securing connected industries*. Birmingham, England: Packt Publishing, 2018.

[4] *CAPE sandbox book — CAPE sandbox v2.1 book*. URL: `https://capev2.readthedocs.io/en/latest/`.

[5] *Debian – The Universal Operating System*. URL: `https://www.debian.org/`.

[6] *Empowering app development for developers*. URL: `https://www.docker.com/`.

[7] Daniel Fraunholz, Marc Zimmermann y Hans D. Schotten. «An adaptive honeypot configuration, deployment and maintenance strategy». En: *International Conference on Advanced Communication Technology, ICACT* (2017), págs. 53-57. ISSN: 17389445.

[8] Van Hauser. *thc-hydra: hydra*. URL: `https://github.com/vanhauser-thc/thc-hydra`.

[9] Muna Al-Hawawreh, Frank Den Hartog y Elena Sitnikova. «Targeted Ransomware: A New Cyber Threat to Edge System of Brownfield Industrial Internet of Things». En: *IEEE Internet of Things Journal* 6.4 (2019), págs. 7137-7151. ISSN: 23274662.

[10] Yan Hu y col. «A survey of intrusion detection on industrial control systems». En: *International Journal of Distributed Sensor Networks* 14.8 (2018). ISSN: 15501477.

[11] Arthur Jicha, Mark Patton y Hsinchun Chen. «SCADA honeypots: An in-depth analysis of Conpot». En: *IEEE International Conference on Intelligence*

and Security Informatics: Cybersecurity and Big Data, ISI 2016 May 2013 (2016), págs. 196-198.

[12] Donald Krambeck. *An Introduction to SCADA Systems - Technical Articles*. 2015. URL: https://www.allaboutcircuits.com/technical-articles/an-introduction-to-scada-systems/.

[13] *Las herramientas Elastic — documentación de ManualKibanaOCDS - latest*. URL: https://manualkibanaocds.readthedocs.io/es/latest/index.html.

[14] Tongbo Luo y col. «IoTCandyJar: Towards an Intelligent-Interaction Honeypot for IoT Devices». En: *Black Hat 2017* (2017), págs. 1-11. ISSN: 2050-7526.

[15] K A Monnappa. *Learning Malware Analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware*. Birmingham, England: Packt Publishing, 2018.

[16] *Overview of Docker Compose*. 2021. URL: https://docs.docker.com/compose/.

[17] Yin Minn Pa Pa y col. «IoTPOT: A novel honeypot for revealing current IoT threats». En: *Journal of Information Processing* 24.3 (2016), págs. 522-533. ISSN: 18826652.

[18] John Park y Steve Mackay. «Titles in the series Practical Cleanrooms: Technologies and Facilities (David Conway) Practical Data Acquisition for Instrumentation and Control Systems». En: (2003), pág. 110.

[19] Niels Provos. «Honeyd: A Virtual Honeypot Daemon». En: *Proceedings of the 10th DFNCERT Workshop* January 2003 (2003), págs. 1-7.

[20] Syed Rizvi y col. «Identifying the attack surface for IoT network». En: *Internet of Things* 9 (2020), pág. 100162. ISSN: 25426605.

[21] Hamdija Sinanovic y Sasa Mrdovic. «Analysis of Mirai malicious software». En: *2017 25th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2017* (2017).

[22] L Spitzner. «Honeypots: catching the insider threat». En: *19th Annual Computer Security Applications Conference, 2003. Proceedings.* 2003, págs. 170-179.

[23] *State of the IoT 2018: Number of IoT devices now at 7B Market accelerating*. 2018. URL: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/.

[24] J Steven Perry. *Anatomía de un ataque de malware a IoT*. IBM Developer. 2019. URL: https://developer.ibm.com/es/articles/iot-anatomy-iot-malware-attack/.

[25] Keith Stouffer y col. *NIST Special Publication 800-82, Guide to Industrial Control Systems (ICS) Security*. Inf. téc. National Institute of Standards y Technology, 2015.

[26] Unit. *2020 unit 42 IoT Threat Report*. 2020. URL: https://unit42.paloaltonetworks.com/iot-threat-report-2020/.

Anexo A

Horas Invertidas

El presente anexo explica el desglose de horas que se han dedicado al desarrollo del trabajo.

A.1. Seguimiento

El seguimiento del trabajo con el tutor se ha llevado a cabo mediante reuniones semanales o bisemanales. En estas reuniones se ha hablado sobre el trabajo llevado a cabo en esas semanas, despejado dudas y realizado planificaciones para las siguientes semanas.

A.2. Desglose de horas

El trabajo desarrollado se divide en las cuatro fases siguientes: *Documentación*; *análisis y diseño*; *desarrollo* y por último, *memoria*. El tiempo que se ha dedicado para su desarrollo ha sido de 503 horas divididas entre las cuatro fases mencionadas anteriormente. La Tabla A.1 y la Figura A.1, muestran el desglose de horas invertidas por fase.

Tabla A.1: Horas invertidas por fase

Fase	Horas invertidas
Documentación	100
Análisis y diseño	126
Desarrollo	201
Memoria	76
Total	**503**

Figura A.1: Horas invertidas por fase. Fuente: propia

Anexo B

Instalación y configuración

El presente anexo detalla los requisitos necesarios para la puesta en marcha de todo el sistema. Esto engloba: requisitos previos, instalación y configuración de la honeynet, instalación y configuración del sandbox y por último, se explica el funcionamiento.

B.1. Requisitos previos

Los requisitos y herramientas necesarias para el sistema son los siguientes:

- Sistema operativo Linux, preferible Ubuntu 18.04 o Debian 10 o distribuciones superiores de cada una.
- Privilegios elevados.
- Sistema actualizado.
- Software de virtualización VirtualBox con una imagen Raspbian i386. También se pueden utilizar VMWare, KVM y XenServer.
- Docker para la creación de contenedores.
- Docker-compose para gestionar de forma sencilla los contenedores.
- Git para realizar un clonado del repositorio.

B.2. Instalación y configuración Honeynet

Para iniciar la honeynet, en primer lugar se debe clonar el repositorio. La instrucción utilizada para ello, es la siguiente:

git clone https://github.com/ibaitxo/iiot-honeynet

Una vez descargado el repositorio, lo siguiente es levantar los contenedores con el comando:

docker-compose up

Ejecutada la instrucción anterior, si es la primera vez que se ejecuta, el usuario debe esperar a que se contruyan las imágenes de los contenedores. La operación puede tardar unos minutos.

Cabe destacar que la honeynet dispone de todos los archivos de configuración necesarios para funcionar, no obstante, de ser necesario, el usuario tiene total libertad de modificarlos.

Por último, se deben configurar los patrones para recoger los logs en Kibana entrando en **localhost:55601**. Posteriormente se debe acceder a la opción de configuración *Manage*. Dentro de *Manage*, en el menú lateral izquierdo, en el menú de Kibana, se debe acceder a *Index Patterns*, donde crearemos un patrón haciendo click sobre *Create Index Pattern*. A continuación, se elige un fichero de registros del menú que se ve en la Figura B.1 y se siguen los pasos. Con los patrones creados, basta con

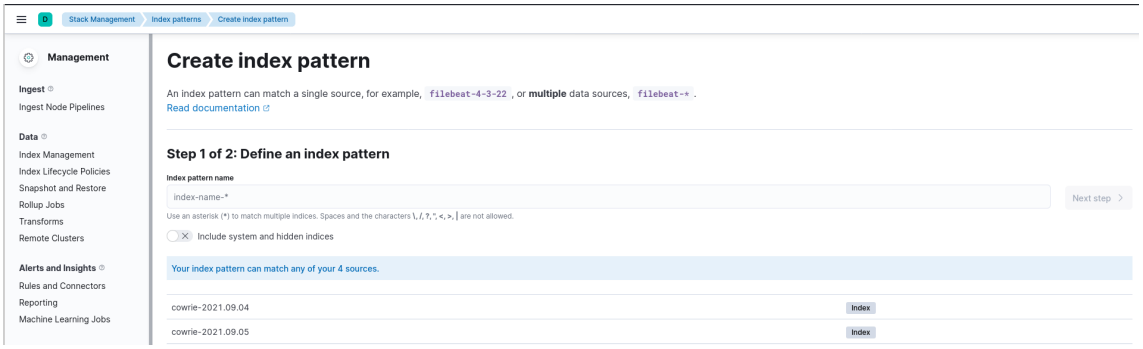

Figura B.1: Creación de patrón de indexado. Fuente: propia

abrir el menú desplegable de la esquina superior izquierda e ir a Analytics - Discover donde se mostrarán los resultados.

También se facilita un script para crear automáticamente las tareas que se realizan en segundo plano y que deben añadirse al planificador de tareas, y otro que es el encargado de gestionar las muestras recogidas y enviarlas a CAPEv2 para ser analizadas.

B.3. Instalación y configuración del sistema de análisis

Para la instalación de CAPEv2, en primer lugar se clona el repositorio:

git clone https://github.com/kevoreilly/CAPEv2

Una vez clonado el repositorio, se proporciona un script con el que ejecutar una isntalación automatizada. En la documentación se avisa que algunas dependencias pueden haber sufrido cambios. Para ejecutar la instalación automatizada se siguen los siguientes pasos:

1. Asignar permisos de ejecución al script: **chmod a+x cape2.sh**
2. Ejecutar el script: **sudo ./cape2.sh base cape**

Con todo instalado, conviene configurar varios ficheros básicos para el funcionamiento, estos son: cuckoo.conf y virtualbox.conf (esto puede cambiar si se utiliza otra máquina virtual). El resto de ficheros son modificables por el usuario. Para cuckoo.conf se deben configurar de la siguiente manera:

- Resultserver
  - ip = 192.168.56.1
  - port = 2042
- Database
  - connection = postgresql://user:password@localhost:54432/cape

En el fichero virtualbox.conf se modifican:

- Virtualbox
  - interface = vboxnet0
  - machines = cuckoo1
- Cuckoo1
  - label = cuckoo1
  - platform = linux
  - ip = 192.168.56.101
  - interface = vboxnet0

Por último, un paso muy importante es configurar la interfaz vboxnet0 de virtualbox. En Archivo - > Administrador de red anfitrión - > Crear aparece una interfaz como la mostrada en la Figura B.2.

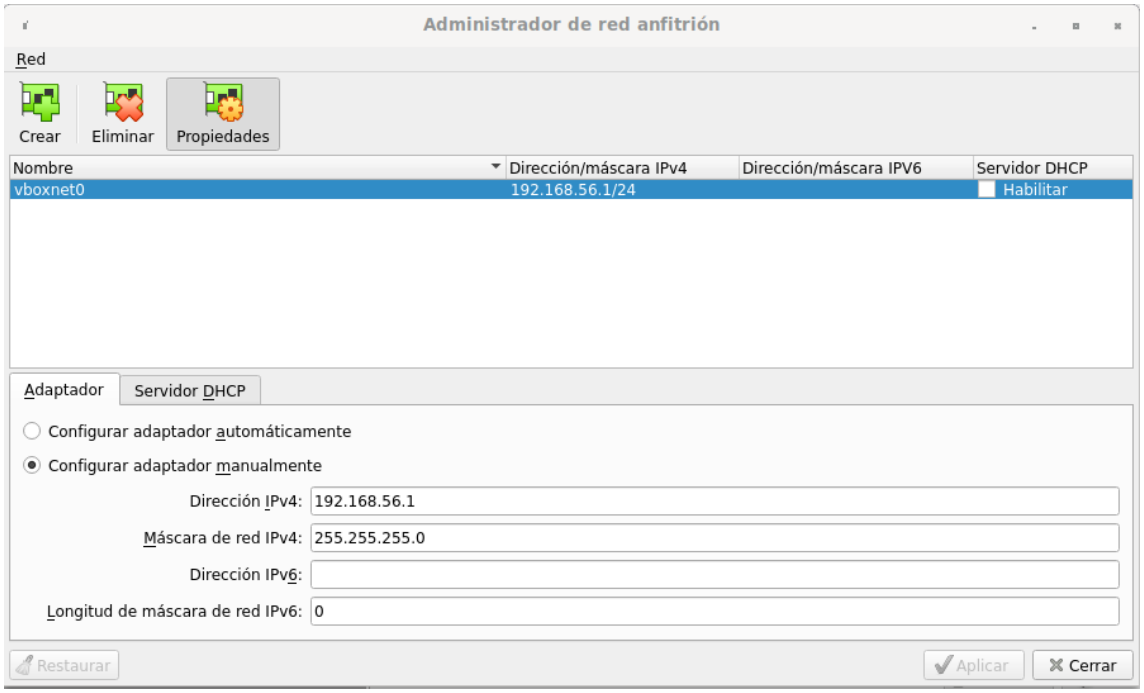

Figura B.2: Interfaz vboxnet0 creada. Fuente: propia

B.4. Funcionamiento

El funcionamiento del sistema es sencillo. Una vez instaladas todas las herramientas y creadas las configuraciones necesarias mencionadas anteriormente, se realizará lo siguiente:

- Levantar los contenedores con la instrucción: ***docker-compose up***
- Crear en ***localhost:5601*** los patrones para visualizar los datos en Kibana con los log recogidos de los contenedores.

Respecto al funcionamiento para el análisis de malware, las funciones están automatizadas. Cada 5 minutos una tarea programada comprueba si existen binarios en la carpeta, envía la muestra (si hay alguna) a CAPEv2 para ser analizada abriendo una VM que realiza el análisis. El reporte que devuelve será almacenado en la carpeta *storage/analysis* dentro de directorio de CAPEv2.

Anexo C

Resultados

En el presente anexo se muestran, a modo de demostración, los resultados obtenidos durante las pruebas realizadas a la honeynet. También se muestra uno de los reportes obtenidos en uno de los análisis de malware realizados.

C.1. Registro de logs

A continuación se muestran los registros de movimientos obtenidos de la honeynet representados a través de Kibana.

La Figura C.1 muestra una traza de logs obtenidos de Honeyd. Analizando la imágen, se aprecia que el formato cumple con el formato de log explicados en la Sección 5.2.

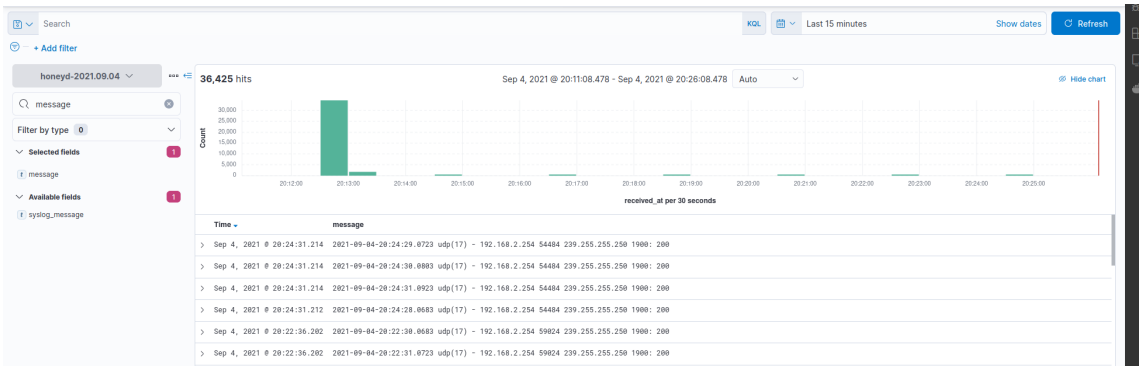

Figura C.1: Muestra de logs de Honeyd. Fuente propia

En la Figura C.2 se muestran los intentos de inicio de sesión recogidos por Cowrie.

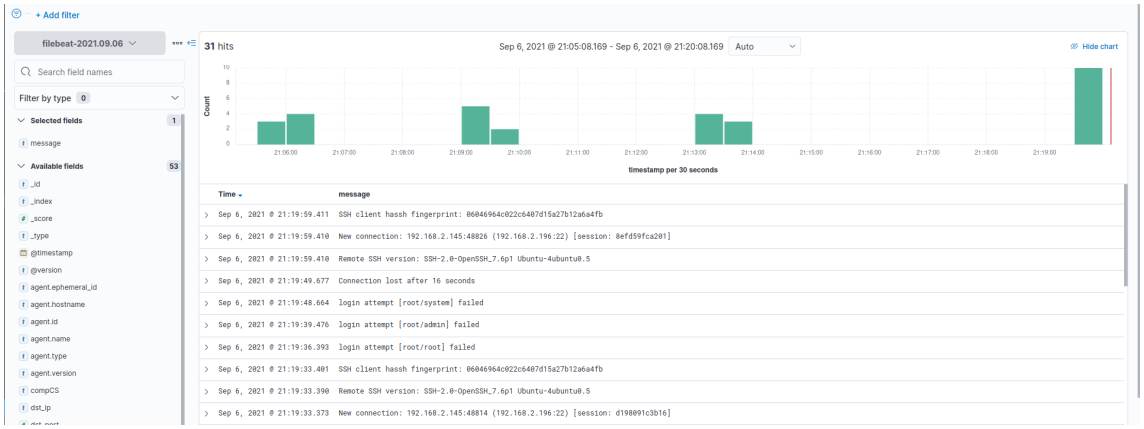

Figura C.2: Muestra de intentos de inicio de sesión en Cowrie. Fuente propia

La Figura C.3 muestra una descarga de software malicioso recogida por Cowrie, y dos intentos de descarga fallidos.

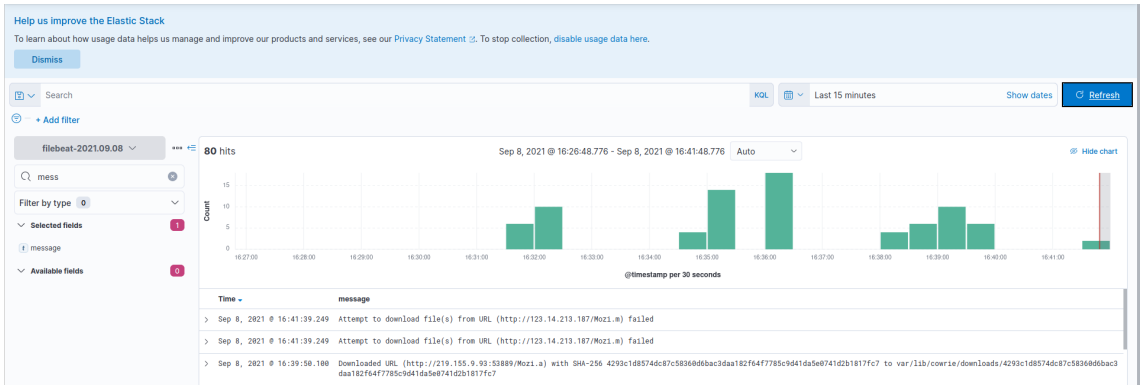

Figura C.3: Muestra de descargas de malware. Fuente propia

C.2. Reporte de análisis

La muestra analizada corresponde al malware Mirai.

La Figura C.4 muestra la información del fichero analizado.

Sandbox Info

Category	Started On	Completed On	Duration	Cuckoo Version
FILE	2021-09-04 16:05:52	2021-09-04 16:06:53	61 seconds	2.2-CAPE

Machine	Label	Manager	Started On	Shutdown On
cuckoo1	cuckoo1	VirtualBox	2021-09-04 16:05:52	2021-09-04 16:06:53

File Details

File name	f60b29cfb7eab3aeb391f46e94d4d8efadde5498583a2f5c71bd8212d8ae92da
File size	283000 bytes
File type	ELF 32-bit LSB executable, ARM, EABI4 version 1 (SYSV), statically linked, with debug_info, not stripped
CRC32	5C0AEF65
MD5	9a6e4b8a6ba5b4f5a408919d2c169d92
SHA1	9b8523cbf0f3af49dbb1680d53c8fc9b2782bcfc
SHA256	f60b29cfb7eab3aeb391f46e94d4d8efadde5498583a2f5c71bd8212d8ae92da
SHA512	d5fd2334772c18729790ec25b5e3c0ace6353aaa853f60d7e55b13f9b88f49e1dec294c303abc3877894ee8a492fdd1d6a0b951405f1f5a021280ff1c1800670
SHA3-384	96cfe6329dea5deb878f338e765d5292eae2b9af44f88c8d1862c2260322c1b267e3f6882aed2f9dcadb379a56e61d7b
TLSH	T1FA54199DC9825E17C4C60B86B6BF92053372533598D767028E04AFB8B7C7A090F36646
Ssdeep	6144:XkYUAmEjloym0V80hkRocENCP0RnYtGSoBmb4d3PCBElKb/0FaiFsXWxATqtEvcM:XkYUAmEjloym0V80hkRo/NCP0RnYtGSj
ClamAV	
Yara	None matched
VirusTotal	VirusTotal lookup disabled, add your API key to the module

Figura C.4: Información del fichero analizado. Fuente propia

En la Figura C.5 muestra las direcciones IP que se han encontrado durante el análisis. Se han encontrado 2962 direcciones, en la imágen solo se muestran algunas.

Signatures

Multiple direct IP connections

direct_ip_connections: Made direct connections to 2962 unique IP addresses

Screenshots

No screenshots available.

Network Analysis

Hosts Involved

Direct	IP Address	Country Name
Y	87.41.105.92	unknown
Y	45.196.130.230	unknown
Y	214.163.132.191	unknown
Y	61.187.200.160	unknown
Y	150.73.157.41	unknown
Y	52.202.130.52	unknown
Y	155.124.166.42	unknown
Y	17.64.96.73	unknown
Y	150.249.3.66	unknown
Y	167.143.6.215	unknown
Y	63.89.120.244	unknown
Y	61.83.80.128	unknown
Y	84.181.23.106	unknown

Figura C.5: Direcciones IP involucradas. Fuente propia

Para terminar, las Figuras C.6 y C.7, muestran el comportamiento del malware y los procesos que haya podido ejecutar. En este caso no se han podido encontrar procesos.

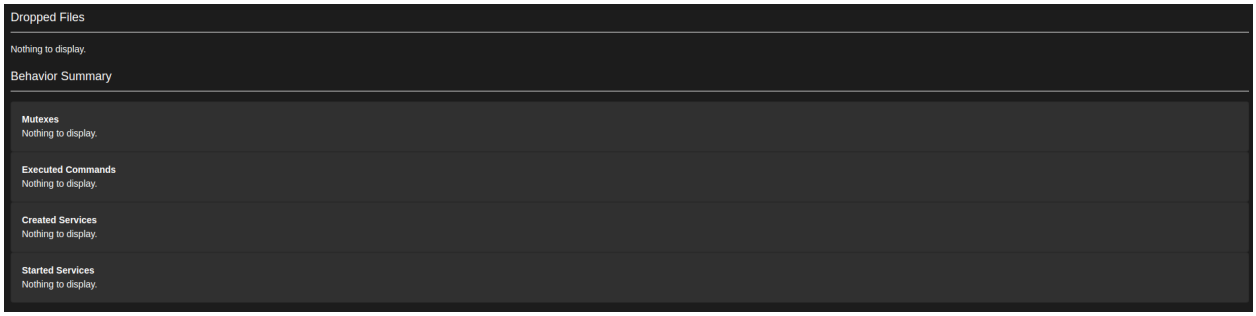

Figura C.6: Comportamiento del malware. Fuente propia

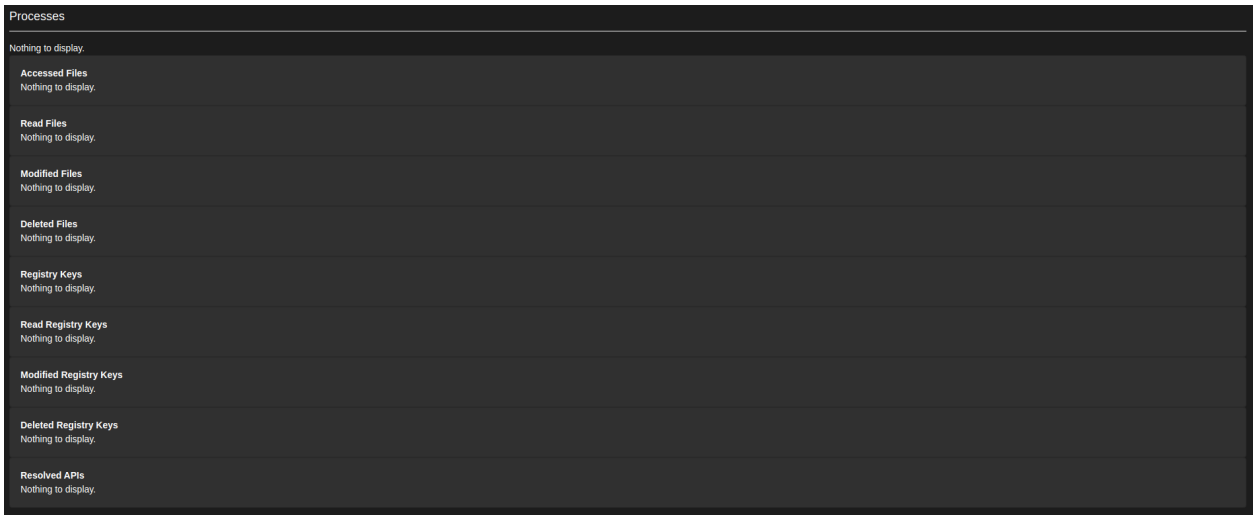

Figura C.7: Procesos que ejecuta el malware. Fuente propia